Data Movement

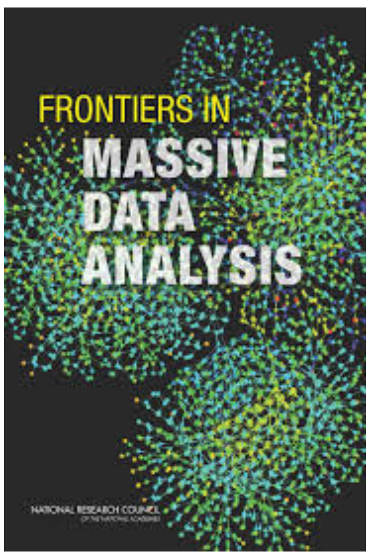
MC F2F

April 2014

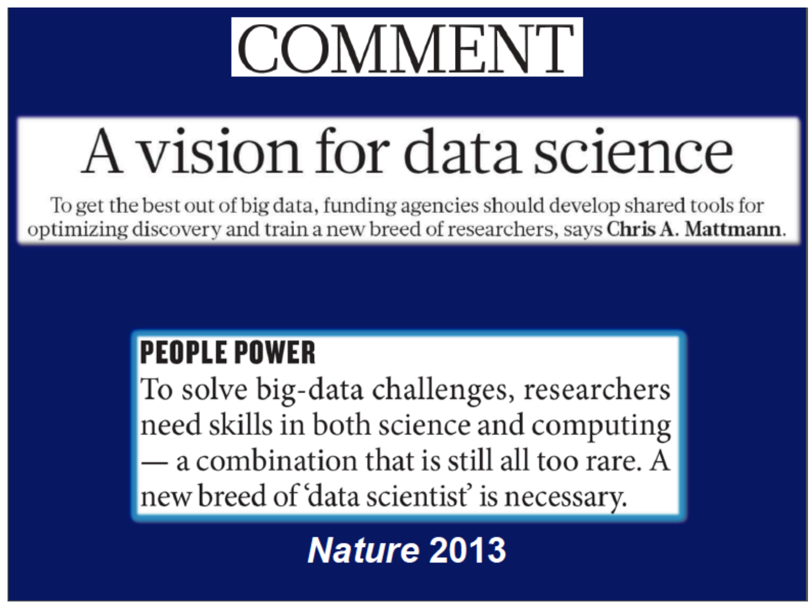
Dan Crichton

Introduction

- Data movement is viewed as one of the fundamental system architecture challenges in scaling data systems



(NRC, Sept 2013)



COMMENT

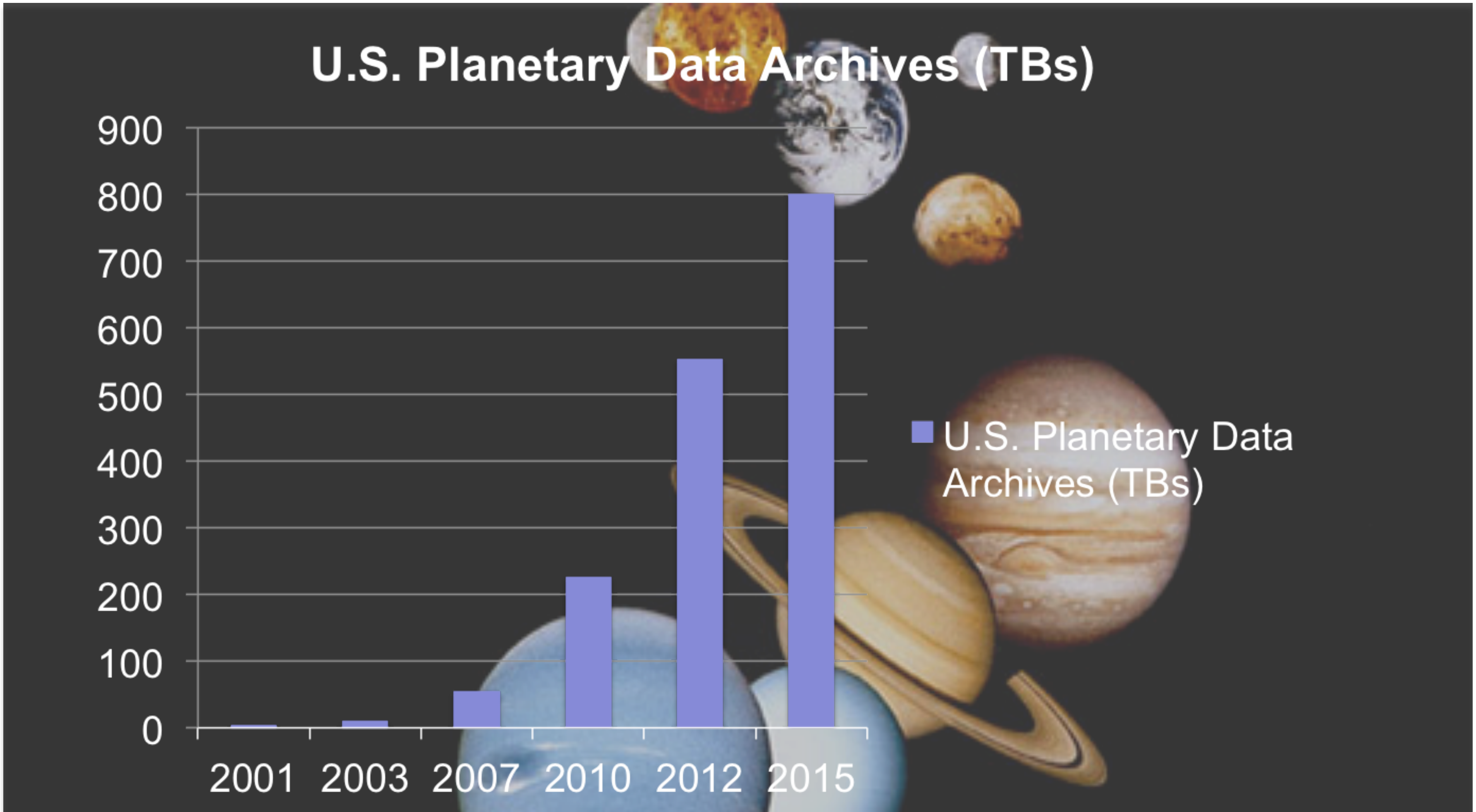
A vision for data science

To get the best out of big data, funding agencies should develop shared tools for optimizing discovery and train a new breed of researchers, says **Chris A. Mattmann**.

PEOPLE POWER
To solve big-data challenges, researchers need skills in both science and computing — a combination that is still all too rare. A new breed of 'data scientist' is necessary.

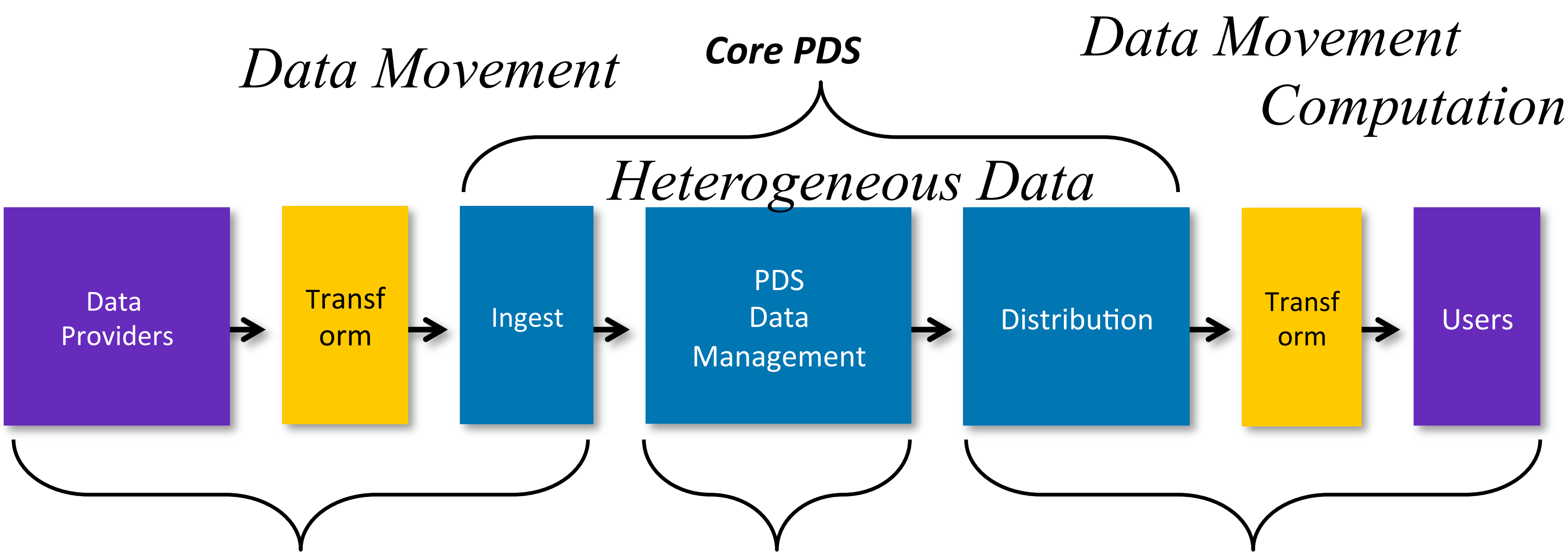
***Nature* 2013**

Growth of Planetary Data Archived from U.S. Solar System Research



Yes, size matters, but so does complexity…

Architecting PDS Towards a Decoupled Architecture



Improve efficiency and support to deliver high quality science products to PDS

Preserve and ensure the stability and integrity of PDS data

Improve user support and usability of the data in the archive

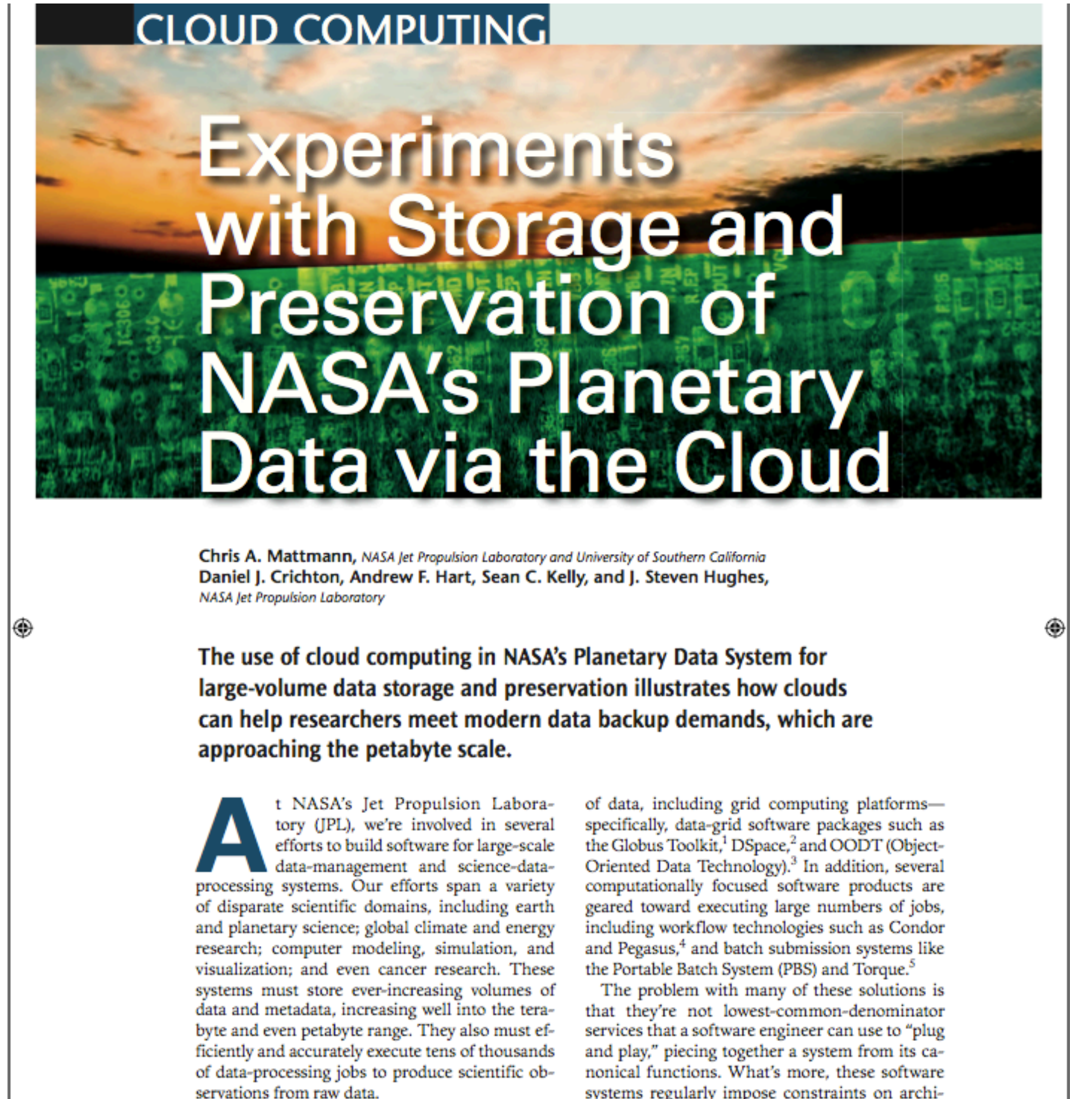
Storage

The Planetary Data Movement Experiment

- Online data movement has been a limiting factor for embracing big data technologies
- Conducted in 2006*, 2009 and 2012
- Evaluate trade offs for moving data
  - to PDS
  - between Nodes
  - to NSSDC/deep archive
  - to Cloud

The Planetary Cloud Experiment

- Utility to PDS
- How does it fit PDS4 architecture
  - APIs
  - Decoupled storage and services
- Data movement challenges
- Cloud Storage as a secondary storage option



CLOUD COMPUTING

Experiments with Storage and Preservation of NASA's Planetary Data via the Cloud

Chris A. Mattmann, *NASA Jet Propulsion Laboratory and University of Southern California*
Daniel J. Crichton, Andrew F. Hart, Sean C. Kelly, and J. Steven Hughes, *NASA Jet Propulsion Laboratory*

The use of cloud computing in NASA's Planetary Data System for large-volume data storage and preservation illustrates how clouds can help researchers meet modern data backup demands, which are approaching the petabyte scale.

At NASA's Jet Propulsion Laboratory (JPL), we're involved in several efforts to build software for large-scale data-management and science-data-processing systems. Our efforts span a variety of disparate scientific domains, including earth and planetary science; global climate and energy research; computer modeling, simulation, and visualization; and even cancer research. These systems must store ever-increasing volumes of data and metadata, increasing well into the terabyte and even petabyte range. They also must efficiently and accurately execute tens of thousands of data-processing jobs to produce scientific observations from raw data.

of data, including grid computing platforms—specifically, data-grid software packages such as the Globus Toolkit,[1] DSpace,[2] and OODT (Object-Oriented Data Technology).[3] In addition, several computationally focused software products are geared toward executing large numbers of jobs, including workflow technologies such as Condor and Pegasus,[4] and batch submission systems like the Portable Batch System (PBS) and Torque.[5]

The problem with many of these solutions is that they're not lowest-common-denominator services that a software engineer can use to "plug and play," piecing together a system from its canonical functions. What's more, these software systems regularly impose constraints on archi-

IEEE Pro, Sept/Oct 2010

Cloud Computing and Computation

- On-demand computation (scaling to massive number of cores)
- Amazon EC2, one of the most popular
- Commoditizing super-computing
- Again, architecting systems to decouple "processing" and "computation" so it can be executed on the cloud is key… two examples
  - This can help scalability for data distribution and science analysis purposes
- Coupled with computational frameworks (e.g., Apache Hadoop)
  - Open source implementation of Map-Reduce

Challenges of Cloud Storage

- Data Integrity
- Ownership (local control, etc)
- Security
- ITAR
- **Data movement to/from cloud**
- Procurement
- Cost arrangements

2012 Data Xfer Technologies Evaluated

- FTP uses a single connection from transferring files; in general it is ubiquitous and where possible the simplest way for PDS to transfer data electronically

- bbFTP uses multiple threads/connections to improve data transfer. It works well as long as the number of connections are kept to a reasonable limit

- GridFTP uses multiple threads/connections. It is part of the Globus project and is used by the climate research community to move models. In general, tests have shown that it is more difficult to set up due to the security infrastructure, etc

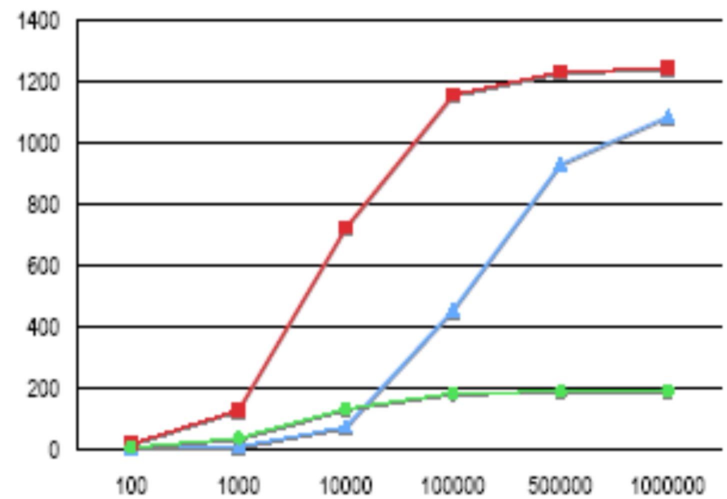
- iRODS uses multiple threads/connections to improve data transfer. It works well as long as the number of connections are kept to a reasonable limit

- FDT uses multiple threads/connections to improve data transfer. It works well as long as the number of connections are kept to a reasonable limit

Some of our Findings

- Transfer speed among the nodes differ greatly, however, the fundamental findings about how to best transfer data for each scenario is consistent
- Parallel transfer mechanisms show improvement over conventional transfer mechanisms (FTP, socket-to-socket) for files larger than ~10MB
- Packaging/bundling small files help to achieve significantly better transfer performance with parallel data transfer
- Reliability has improved over the past five years in many of the products we have tested
  - However, UDP approaches have suffered largely due to more aggressive network infrastructure seeing this as distributed denial of service attacks (DDOS)

Data Movement of WAN using TCP/IP



Transfer rate (Y axis) versus file size (X axis)
GridFTP: blue, bbFTP: red, FTP: green

Data Movement Trade

	FTP	bbFTP	GridFTP	Data Brick	FDT	iRODS
Efficiency	High for files < 1 GB	High	Slightly lower than bbFTP	Low	Very High	High
Scalability	Linear	Based on number of threads	Based on number of threads	Based on available storage sizes	Adaptive	Adaptive
Reliability	Fault rate dependent on underlying TCP/IP protocol, but 0 faults / 20 hours of testing and 10s of GBs of data	Good (support retransmit, issue with > 12 threads)	High (support retransmit)	High	Poor	Excellent
Ease of Use	Easy	Easy	Medium	Based on brand	Medium	Easy
Ease of Deployment	Easy (standard component on Linux/ UNIX/Mac, and some Windows solutions)	Easy to deploy on Unix based systems with / etc/passwd security. Can also use Globus GSI security)	Difficult to deploy; relies on Grid Security Infrastructure and certificate management for hosts, users, services	Based on brand	Medium	Difficult
Cost (Operate & Implement)	Low	Low	Medium (hard to deploy)	Based on brand & volume	Low	Low

Pilot with DNs

- Focus on movement between nodes (including mission nodes) and NSSDC
  - Distribution of massive data vs. subsetting data for user access is something that still may need to be addressed
- iRODS has shown to be the most promising for data transfer
  - Geo: 1.2 MiB/s FTP, 8.8 MiB/s iRODS
  - IMG-JPL: 1.1 MiB/S FTP, 5.8 MiB/s iRODS
  - Note, there are lots of groups doing different things these days (e.g., ESGF, CERN, etc)
- Setting up an iRODS infrastructure for data movement with 4 zones: GEO, USGS, JPL/IMG, and NSSDC as a pilot

Status

- iRODS Servers:
  - Geo node - operational
  - PDS Imaging (JPL)- operational
- iRODS Clients:
  - NSSDC - installation pending
- File Transfers:
  - Functional & accurate
  - Improved speed noticed with iRODS versus conventional FTP, HTTP
  - Improvements dependent on file sizes, composition, time of day
- Documentation:
  - Pilot project plan: https://oodt.jpl.nasa.gov/wiki/x/FADFAg
  - Server setup guide:https://oodt.jpl.nasa.gov/wiki/x/EICnAg
  - Client setup guide: https://oodt.jpl.nasa.gov/wiki/x/cgDxAg
  - Users guide: https://oodt.jpl.nasa.gov/wiki/x/PQDZAg
  - Server list: https://oodt.jpl.nasa.gov/wiki/x/dgDxAg

Benchmarks

JPL to Geo

Technology	File Size 1 MiB	10 MiB	100 MiB	1 GiB	2 GiB
TCP 1	0.55	0.94	0.93	1.33	0.94
TCP 2	0.55	1.07	2.58	2.68	2.73
TCP 4	0.55	1.19	5.07	5.46	5.45
TCP 8	0.56	1.19	8.95	10.6	10.79
TCP 16	0.56	1.19	12.02	18.45	20.32

Geo to JPL

Technology	File Size 1 MiB	10 MiB	100 MiB	1 GiB	2 GiB
TCP 1	0.36	0.61	0.66	0.58	0.68
TCP 2	0.36	0.63	1.31	1.36	1.37
TCP 4	0.39	0.62	2.26	2.69	2.7
TCP 8	0.41	0.62	3.8	5.06	5.2
TCP 16	0.41	0.63	5.72	8.06	8.87

Benchmarks (2)

USGS to JPL

Technology	File Size 1 MiB	10 MiB	100 MiB	1 GiB	2 GiB
TCP 1	1.29	2.11	2.59	2.61	1.78
TCP 2	0.93	2.59	3.6	4.01	2.6
TCP 4	0.9	1.87	4.3	4.17	3.22
TCP 8	0.89	2.56	3.95	4.28	3.86
TCP 16	0.89	2.16	4.16	4.19	3.84

JPL to USGS

Technology	File Size 1 MiB	10 MiB	100 MiB	1 GiB	2 GiB
TCP 1	0.87	0.89	0.88	0.96	N/A
TCP 2	0.83	1.01	1.71	1.81	N/A
TCP 4	0.77	0.91	2.45	3.03	3.12
TCP 8	0.87	1.02	2.89	3.73	3.76
TCP 16	0.81	0.74	3.55	3.79	4.02

LBNL Collaboration

- LBNL runs the *es.net* center which evaluates data movement capabilities for various virtual science networks (climate, energy, etc)

- Eli Dart (LBNL, es.net) approached Dan after a presentation and asked about a collaboration
  - They may be willing to evaluate our data movement capabilities in PDS to tell us how we compare

Next steps

- GEO and EN collaborating on a real life scenario
- Complete NSSDC setup and evaluate push/pull models for online transfer and delivery